SELP: Semi-supervised evidential label propagation algorithm for graph data clustering

Kuang Zhou[a,b,*], Arnaud Martin[c], Quan Pan[b], Zhunga Liu[b]

[a]*School of Natural and Applied Sciences, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China*

[b]*School of Automation, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China*

[c]*DRUID, IRISA, University of Rennes 1, Rue E. Branly, 22300 Lannion, France*

Abstract

With the increasing size of social networks in the real world, community detection approaches should be fast and accurate. The label propagation algorithm is known to be one of the near-linear solutions which is easy to implement. However, it is not stable and it cannot take advantage of the prior information about the network structure which is very common in real applications. In this paper, a new Semi-supervised clustering approach based on an Evidential Label Propagation strategy (SELP) is proposed to incorporate limited domain knowledge into the community detection model. The main advantage of SELP is that it can effectively use limited supervised information to guide the detection process. The prior information about the labels of nodes in the graph, including the labeled nodes and the unlabeled ones, is initially expressed in the form of mass functions. Then the evidential label propagation rule is designed to propagate the labels from the labeled nodes to the unlabeled ones. The communities of each node can be identified after the propagation process becomes stable. The outliers can be identified to be in a special class. Experimental results demonstrate the effectiveness of SELP on both graphs and classical data sets.

Keywords: Semi-supervised learning, Label propagation, Theory of belief functions, Uncertainty, Community detection

1. Introduction

As described in [2], communities (also called clusters or modules) are groups of nodes (vertices) which probably share common properties and/or play similar roles within the graph (or network)[1]. Identifying communities may offer insight on how the network is organized [3], and it is often the precondition for the structural and functional analysis of the networked systems. Community detection for networks has attracted considerable attention crossing many areas from physics, biology, and economics to sociology [3]. It can be seen as the task of clustering on graph data, which consists of a finite set of nodes, together with a set of

*Corresponding author.

Email addresses: kzhoumath@163.com (Kuang Zhou), Arnaud.Martin@univ-rennes1.fr (Arnaud Martin), quanpan@nwpu.edu.cn (Quan Pan), liuzhunga@nwpu.edu.cn (Zhunga Liu)

This paper is an extension and revision of [1].

[1]In this work, "graph" and "network" are considered as synonyms.

unordered pairs of these vertices. These pairs are known as edges in the graph.

As the size of real-world networks grows rapidly, the community detection algorithms need to be fast and efficient. The Label Propagation Algorithm (LPA), which was first investigated by Raghavan et al. [4], has the benefits of nearly-linear running time and easy implementation. But the original LPA is not stable due to randomness. Different communities may be detected in different runs over the same network. Moreover, by assuming that a node always adopts the label of the majority of its neighbors, LPA ignores any other structural information existing in the neighborhood. In real applications, there is often some prior information about the network structure. For instance, in co-authorship networks, the communities related to some famous scholars are easy to know. In the movie network, the types of some special films may be clear to us. If such kind of prior information could be fused effectively in the unsupervised community detection models, the performance could be improved.

Supervised classification is one of the most popular techniques in machine learning. Generally, the goal of supervised learning is to train a classifier that reliably approximates a classification task based on a set of labeled examples from the problem of interest. The performance of the learned classifier highly depends on the proportion of labeled samples. However, in many practical applications of pattern classification, it is usually difficult to get abundant labeled samples since the task of manual labeling is time consuming and often requires expensive human labor. On the contrary, there are usually a large number of unlabeled samples which are easier to obtain. Consequently, Semi-Supervised Learning (SSL), which aims to effectively combine the unlabeled data with labeled data, has been developed to perform the classification task when there are not enough training data.

Some semi-supervised community detection approaches have already be proposed [5–7]. The supervised information in these models are mainly two types: 1. The labels of some nodes are given in advance; 2. There are some must-link and/or cannot-link pair-wise constrains between some node pairs. In this paper we focus on the former type, *i.e.,* some nodes in the graph are assumed to be labeled in advance. There are some problems when dealing with the information about node labels among the existing semi-supervised community detection methods, such as:

- If there are some outliers in the graph, the performance of the community detection model will become worse.

- If the labeled objects are located in the overlapping region between or among communities, the same label will be propagated to more than one class and, consequently, the accuracy of the detection results will be low.

The theory of belief functions is very effective in dealing with uncertain information, and it has already been applied in many fields, such as data classification [8–12], data clustering [13–16], complex networks [17–20], data fusion [21] and statistical estimation [22–24]. In this work, we try to address the above problems in semi-supervised community detection models using the theory of belief functions. The Semi-supervised Evidential Label Propagation (SELP) algorithm will be proposed to take advantage of the prior information in the graph. The

initial knowledge about node labels is expressed in the form of Bayesian categorical mass functions, while the labels of the unlabeled nodes are represented by vacuous mass functions. The evidential label propagation rule is designed to propagate the labels from the labeled nodes to the unlabeled ones iteratively. The basic belief assignments about each nodes' classes are obtained after convergence of the algorithm. Experimental results show that SELP can improve the accuracy of the detected communities compared with the unsupervised version. This result confirms that limited supervised information is of great value for the community detection task.

The rest of this paper is organized as follows. In Section 2, some basic knowledge and the rationale of our method are briefly introduced. In Section 3, the proposed SELP algorithm will be presented in detail. In order to show the effectiveness of the proposed community detection approaches, in Section 4 we test the SELP algorithm on different artificial and real-world data sets and compare it with related partitioning methods. Finally, we conclude and present some perspectives in Section 5.

2. Background

In this section some related preliminary knowledge, including the theory of belief functions and the classical label propagation algorithm, will be presented.

2.1. Theory of belief functions

Let $\Omega = \{\omega_1, \omega_2, \ldots, \omega_c\}$ be the finite domain of X, called the discernment frame. The belief functions are defined on the power set 2^Ω. Function $m : 2^\Omega \to [0, 1]$ is said to be a Basic Belief Assignment (bba) on 2^Ω, if it satisfies:

$$\sum_{A \subseteq \Omega} m(A) = 1. \tag{1}$$

Every $A \in 2^\Omega$ such that $m(A) > 0$ is called a focal element. The credibility and plausibility functions can be defined, respectively, as

$$Bel(A) = \sum_{B \subseteq A, B \neq \emptyset} m(B) \ \ \forall A \subseteq \Omega, \tag{2}$$

$$Pl(A) = \sum_{B \cap A \neq \emptyset} m(B), \ \ \forall A \subseteq \Omega. \tag{3}$$

Each quantity $Bel(A)$ measures the total support given to A, while $Pl(A)$ represents potential amount of support to A. Based on the types of the focal set that contains all the focal elements, we can have some particular forms of mass functions. A *categorical mass function* is a normalized bba which has a unique focal element A^*. This kind of mass functions can be defined as:

$$m(A) = \begin{cases} 1 & \text{if } A = A^* \subset \Omega, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

A *vacuous mass function* is a particular categorical mass function focused on Ω. It is a special kind of categorical mass functions with a unique focal element Ω. This type of mass functions is defined as follows:

$$m(A) = \begin{cases} 1 & \text{if } A = \Omega, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

The vacuous mass function represents the case of total ignorance. A *Bayesian mass function* is a bba whose all focal elements are elementary hypotheses (*i.e.,* singletons). It can be regarded as a probability distribution over frame Ω. Specially, if a Bayesian mass function is categorical, it represents that there is no uncertainty at all and we are completely sure about the state of the concerned variable.

A belief function can be transformed into a probability function by Smets' method [25], in which the mass $m(A)$ is equally distributed among the elements of A. This leads to the concept of pignistic probability, $BetP$, defined by

$$BetP(\omega_i) = \sum_{\omega_i \in A \subseteq \Omega} \frac{m(A)}{|A|(1 - m(\emptyset))}, \tag{6}$$

where $|\cdot|$ denotes the number of elements of the set.

Mass functions can be used to describe the information obtained from different sources. It requires to take into account the level of reliability of each information source in real applications. When the sources of evidence are not completely reliable, the discounting operation proposed by Shafer [26] and justified by Smets [27] can be applied. Denote the reliability degree of mass function m by $\alpha \in [0, 1]$, then the discounting operation can be defined as:

$$m^{'}(A) = \begin{cases} \alpha \times m(A) & \forall A \subset \Theta, \\ 1 - \alpha + \alpha \times m(\Theta) & \text{if } A = \Theta. \end{cases} \tag{7}$$

If $\alpha = 1$, the evidence is completely reliable and the bba will remain unchanged. On the contrary, if $\alpha = 0$, the evidence is completely unreliable. In this case the so-called vacuous belief function, $m(\Theta) = 1$, is obtained. It describes our total ignorance.

How to combine efficiently several bbas coming from distinct sources is a major information fusion problem in the theory belief functions. Many rules have been proposed for such a task. When the information sources are considered as reliable, several distinct bodies of evidence characterized by different bbas can be combined using Dempster-Shafer (DS) rule [26]. If bbas $m_j, j = 1, 2, \cdots, S$ describing S distinct items of evidence on Ω, the combined

bba of the S mass functions using the DS rule can be obtained by the following equation:

$$(m_1 \oplus m_2 \oplus \cdots \oplus m_S)(X) = \begin{cases} 0 & \text{if } X = \emptyset, \\ \dfrac{\sum\limits_{Y_1 \cap \cdots \cap Y_S = X} \prod\limits_{j=1}^{S} m_j(Y_j)}{1 - \sum\limits_{Y_1 \cap \cdots \cap Y_S = \emptyset} \prod\limits_{j=1}^{S} m_j(Y_j)} & \text{otherwise.} \end{cases} \tag{8}$$

A decision can finally be made by assigning object o_i to the class ω_k with the highest plausibility.

2.2. EK-NNclus clustering

Belief functions defined on the power set can well describe the uncertainty in the class structure in the analyzed data set. Many clustering algorithms have been designed using the theory of belief functions [13, 14, 16, 28, 29]. See Ref.[30] for a review. Recently, Denœux et al. [31] put forward a new decision-directed clustering algorithm, named EK-NNclus, which uses the evidential K nearest-neighbor (EK-NN) rule [8] as the base classifier.

EK-NNclus is a clustering algorithm for relational data, where the dissimilarities between objects should be given in advance. Consider a clustering problem for n objects. The dissimilarity between objects o_i and o_j is denoted by $d_{ij}, i, j = 1, 2, \cdots, n$. Matrix $D = (d_{ij})$ is symmetric. Let $\Omega = \{\omega_1, \cdots, \omega_c\}$ be the set of groups. Define class-membership binary variables u_{ik}, which indicates whether object o_i belongs to cluster ω_k ($u_{ik} = 1$) or not ($u_{ik} = 0$). To initialize the algorithm, the objects can be labeled randomly (or using some prior knowledge if available). As the number of clusters is unknown, we can simply set $c = n$, *i.e.* it is assumed that there are as many clusters as objects and each cluster contains exactly one object. The algorithm iteratively reassigns objects to clusters in some random order using the EKNN rule. For each object o_i, the knowledge that object o_t is at a distance d_{it} from o_i is a piece of evidence for the class membership of o_i, which can be represented by the following mass function defined on Ω:

$$\begin{cases} m_t^i(\{\omega_k\}) = u_{tk}\alpha_0 \exp\{-\gamma d_{it}\}, \quad k = 1, 2, \cdots, c \\ m_t^i(\Omega) = 1 - \alpha_0 \exp\{-\gamma d_{it}\}, \end{cases} \tag{9}$$

where α_0 and γ are some constants. Denoting by N_i^K the set of the K nearest neighbors of object o_i, the K mass functions $m_t^i, t \in N_i^K$ can be combined by the DS rule:

$$m^i = \bigoplus_{t \in N_i^k} m_t^i. \tag{10}$$

The label of object o_i can be determined according to the fused mass function m^i. As the focal elements of m^i are the singletons and the whole frame Ω, the object can be assigned to the cluster with the highest mass assignment (or plausibility). If the label of at least one object has been changed during the last iteration, then the objects are randomly re-ordered and a new iteration is started. Otherwise, the algorithm stops. After convergence, the cluster

membership of each object is described by a mass function assigning a mass to each specific cluster and to the whole set of clusters. One of the advantages of EK-NNclus clustering is that it does not require the number of clusters to be fixed in advance [31].

2.3. Label propagation

The graph data can be denoted by $G(V, E)$, where V is the set of nodes and E is the set of edges. The number of nodes in graph $G(V, E)$ is denoted by $|V|$. It is assumed that each node $n_v(\in V)$ has a label $y_v \in \Omega$, where $\Omega = \{\omega_1, \omega_2, \cdots, \omega_c\}$ represents the frame of classes. Denote by N_v the set of neighbors of node n_v. The Label Propagation Algorithm (LPA) uses the network structure alone to guide its process. It starts from an initial configuration where every node has a unique label. Then at every step one node (in asynchronous version) or each node (in a synchronous version) updates its current label to the label shared by the maximum number of its neighbors. For node n_v, its new label can be updated to ω_j with

$$\omega_j = \arg\max_{\omega_l} \{|n_u : y_u = \omega_l, n_u \in N_v|\}. \tag{11}$$

When there are multiple maximal labels among the neighbors' labels, the new label is picked randomly from them. By this iterative process densely connected groups of nodes form consensus on one label to form communities, and each node has more neighbors in its own community than in any of other community. Communities are identified as a group of nodes sharing the same label.

2.4. Semi-supervised label propagation

The original LPA can only make use of the network topology information, while it completely ignores the background information of the networks. However, in many real-world applications, there may exist some prior information that can be useful in detecting the community structures. Liu et al. [5] proposed a novel semi-supervised community detection approach based on label propagation (SLP), which can utilize the limited prior information to guide the discovery process of community structure.

Let $G(V, E)$ denote the graph, and $\boldsymbol{A} = (a_{ij})_{|V|\times|V|}$ be the adjacent matrix, where $a_{ij} = 1$ if there is an edge between nodes n_i and n_j, and 0 otherwise. Suppose that a node n_v carries a label denoting the community to which it belongs, then n_v propagates its label to its neighbors $n_{v_1}, n_{v_2}, \cdots, n_{v_{|N_v|}}$, where N_v is the set of all the neighbors of node n_v and $|N_v|$ is the degree of node n_v. The node n_{v_i}absorbs a fraction of label information from its neighborhood node n_v, and retains some label information of its initial state. A few number of nodes can be initialized based on the available prior information and then let the labels propagate through the network.

Suppose there are k communities $(\omega_1, \omega_2, \cdots, \omega_k)$ and let $\mathcal{F}$ denote a set of $n \times k$ matrices with non-negative real-value entries. Any matrix $F = [f_1, f_2, \cdots, f_n]^T \in \mathcal{F}$ corresponds to a specific partition on V. Here f_i is a column vector with length k, and the i^{th} element in f_i denotes the membership of node n_i to community ω_i. Initially, we set $F_0 = Y$, where $Y_{ij} = 1$ if node n_i is labeled as ω_j, and $Y_{ij} = 0$ otherwise. For unlabeled nodes $Y_j = 0 (1 < j < k)$.

Then the iteration equation can be given as

$$F^{t+1} = \alpha W F^t + (1-\alpha)Y, \tag{12}$$

where W is the weight matrix with entries

$$w_{ij} = \frac{a_{ij}}{|N_i|}. \tag{13}$$

After convergence, each node can be assigned to the class with the highest membership value, *i.e.,* we can label $n_i \in V$ as

$$y_i = \arg\max_j F_{ij}. \tag{14}$$

3. Semi-supervised evidential label propagation

Inspired from LPA and EK-NNclus [31], we propose here the SELP algorithm for graphs with some prior information. The problem of semi-supervised community detection will be first described in a mathematical way, and then the proposed semi-supervised label propagation algorithm will be presented in detail.

3.1. Problem restatement and notions

As before, let $G(V, E)$ denote the graph, where V is the set of nodes and $E \subseteq V \times V$ is the set of edges. The adjacent matrix of the graph can be denoted by $\boldsymbol{A} = (a_{ij})_{|V|\times|V|}$.

Assume that there are c communities (*i.e.,* clusters, groups) in the graph. The set of labels is denoted by $\Omega = \{\omega_1, \omega_2, \cdots, \omega_c\}$. Each node $n_v \in V$ in the graph is assumed to have a label $\omega_v \in \Omega$. In addition, in order to make sure the solution is unique, we assume that there must be at least one labeled vertex in each community. The $|V|$ nodes in set V can be divided into two parts:

$$V_L = \{n_1, n_2, \cdots, n_l\}$$

for the labeled nodes, and

$$V_U = \{n_{l+1}, n_{l+2}, \cdots, n_{|V|}\}$$

for the unlabeled ones. The nodes in V_L are labeled. Denote the label of node $n_i (\in V_L)$ is $y_i \in \Omega$. The main task of the semi-supervised community detection is to propagate the labels from nodes in V_L to those in V_U, and further determine the labels of those unlabeled vertices.

3.2. Dissimilarities between nodes

A basic assumption which has been often adopted in semi-supervised graph-based learning methods is that nearby points are likely to have the same labels, which is known as the smoothness assumption [32]. Pairwise similarity measure is the basis of label propagation. Similarly, here we assume that the more common neighbors the two nodes share, the larger the

probability that they belong to the same community. Thus, in this work, an index considering the number of shared common neighbors is adopted to measure the similarities between nodes.

Definition 1. If there is an edge between node n_i and n_j, *i.e.*, $a_{ij} = 1$, we say that n_i (n_j) is a neighbor of n_j (correspondingly, n_i). Let the set of neighbors of node n_i be N_i, and the degree of node n_i be $|N_i|$, $i = 1, 2, \cdots, |V|$. The similarity between nodes n_i and n_j ($n_i, n_j \in V$) can be defined as

$$s_{ij} = \begin{cases} \frac{|N_i \cap N_j|}{|N_i| + |N_j|}, & \text{if } a_{ij} = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

Then the dissimilarities associated with the similarity measure can be defined as

$$d_{ij} = \frac{1 - s_{ij}}{s_{ij}}, \ \forall \, n_i, n_j \in V. \tag{16}$$

3.3. Evidential label propagation

For the labeled node $n_j \in V_L$ in community ω_k, the initial bba can be defined as a Bayesian categorical mass function:

$$m^j(A) = \begin{cases} 1 & \text{if } A = \{\omega_k\}, \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

For the unlabeled node $n_x \in V_U$, the vacuous mass assignment can be used to express our ignorance about its community label:

$$m^x(A) = \begin{cases} 1 & \text{if } A = \Omega, \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

To determine the label of node n_x, its neighbors can be regarded as distinct information sources. If there are $|N_x|$ neighbors for node n_x[2], the number of sources is $|N_x|$. Based on the assumption that similar nodes are likely to have the same labels, the reliability of each source depends on the similarities between the neighbors and node n_x. Suppose that there is a neighbor n_t with label ω_j, it can provide us with a bba describing the belief on the community label of node n_x as

$$\begin{aligned} m_t^x(\{\omega_j\}) &= \alpha * m^t(\{\omega_j\}), \\ m_t^x(\Omega) &= m^t(\Omega) + (1 - \alpha) * m^t\left(\{\omega_j\}\right), \\ m_t^x(A) &= 0, \ \text{ if } A \neq \{\omega_j\}, \Omega, \end{aligned} \tag{19}$$

where α is the discounting parameter such that $0 \leq \alpha \leq 1$. It should be determined according to the similarity between nodes n_x and n_t. The more similar the two nodes are, the more reliable the source is. Thus α can be set as a decreasing function of d_{xt}. In this work we

[2]The number of node n_x's neighbors is determined by the number of nodes sharing the same edge with n_x.

suggest to use

$$\alpha = \alpha_0 \exp\left\{-\gamma d_{xt}^{\beta}\right\}, \tag{20}$$

where parameters α_0 and β can be set to be 1 and 2 respectively as default, and γ can be set to

$$\gamma = 1/\text{median}\left(\left\{d_{ij}^{\beta},\ i = 1, 2, \cdots, n,\ j \in N_i\right\}\right). \tag{21}$$

After the $|N_x|$ bbas from its neighbors are calculated using Eq. (19), the fused bba of node n_x can be got by the use of Dempster's combination rule (see Eq. (8)):

$$m^x = m_1^x \oplus m_2^x \oplus \cdots \oplus m_{|N_x|}^x. \tag{22}$$

As all the bbas to combine are in the form of Eq. (19), the focal elements of m^x are the singletons $\{\omega_1\}, \{\omega_2\}, \cdots, \{\omega_c\}$ and the frame Ω. The label of node n_x can be assigned to the focal element with the maximal mass value in m^x. The main principle of semi-supervised learning is to take advantage of the unlabeled data. It is an intuitive way to add node n_x (previously in set V_U but already be labeled now) to set V_L to train the classifier. However, if the predicted label of n_x is incorrect, this will have very bad effects on the accuracy of the following predictions. Here a parameter η is introduced to control the prediction confidence of the nodes that are to be added in V_L. If the maximum of m^x is larger than η, it indicates that the belief about the community of node n_x is high and the prediction is confident. Then we remove node n_x from V_U and add it to set V_L. On the contrary, if the maximum of m^x is not larger than η, it means that we can not make a confident decision about the label of n_x based on the current information. Thus the node n_x should be remained in set V_U. This is the idea of self-training [33], which attempts to iteratively choose to label several examples about which one is the most confident in the unlabeled set. The learner keeps on labeling unlabeled examples and retraining itself on an enlarged labeled training set until all the samples are labeled.

In order to propagate the labels from the labeled nodes to the unlabeled ones in the graph, a classifier should first be trained using the labeled data in V_L. For each node n_x in V_U, we find its direct neighbors and construct bbas through Eq. (19). Then the fused bba about the community label of node n_x is calculated by Eq. (22). The subset of the unlabeled nodes, the maximal bba of which is larger than the given threshold η, are selected to augment the labeled data set. The predicted labels of these nodes are set to be the class assigned with the maximal mass. Parameter η can be set to 0.7 by default in practice.

After the above update process, there may still be some nodes in V_U. For these nodes, we can find their neighbors that are in V_L, and then use Eqs. (19) and (22) to determine their bbas. The whole algorithm of SELP is summarized in Algorithm 1.

The main idea of SELP is similar to that of EK-NNclus clustering algorithm. Both methods update the labels of objects iteratively based on the neighborhood information. The differences between SELP and EK-NNclus algorithms are as follows:

- SELP is in line with the principle of semi-supervised learning, which can take advantage the limited supervised information; EK-NNclus is a clustering method, which is in the

scope of unsupervised learning;

- SELP is specially designed for the community detection task on graphs, while EK-NNclus is for relational data. When using SELP, the dissimilarities between nodes are calculated based on the graph structure;

- In the iterative label propagation process, the updating rule of SELP considers the mass assignments for the neighbors. The dissimilarities between nodes are used to determine the discounting factor.

Algorithm 1 : SELP algorithm

Input: Graph $G(V, E)$. The set of labeled nodes V_L, and the set of unlabeled nodes V_U.
Parameters:
η: the parameter to control the prediction confidence
α_0, β: the parameter to determine the discounting factor
$MaxIts$: the maximal update steps
$PercFul$: the percentage of the labeled data
Initialization:
(1) Initialize the bba of each node in the network using Eqs. (17) and (18).
(2) Let $it = 0$
repeat
(1) For each node $n_x \in V_U$, find the set N_x of all its neighbors and construct $|N_x|$ bbas using Eq. (19).
(2) Calculate the fused bba of node n_x by Eq. (22).
(3) If the maximum of mass assignment of n_x is larger than the threshold η, move node n_x from set V_U to set V_L.
(4) $it = it + 1$.
until The percentage of nodes in V_L is larger than $PercFul$ or the maximal update step is reached.
If there are still some nodes in V_U, update their bbas based on the information from the neighbors using Eqs. (19) and (22).
Output: The bba matrix $\boldsymbol{M} = \{m^i\}$, $i = 1, 2, \cdots, |V|$.

3.4. Application on relational data

To apply the SELP algorithm on classical data sets, an appropriate graph should be constructed first to model the analyzed data set. However, in graph-based learning methods, the construction of graph has not been studied extensively [34, 35]. In this paper, a commonly used method, the K-Nearest Neighbor Graph (KNNG), is adopted to construct a graph based on the dissimilarities between objects [36]. Assume that there are $|V|$ objects, $x_1, x_2, \cdots, x_{|V|}$, in the data set. The dissimilarities between objects are denoted by $d_{ij}, i, j = 1, 2, \cdots, |V|$. The corresponding KNNG, denoted by $G(V, E)$, can be defined based on the dissimilarities as follows. The data points are used as vertices in the constructed undirected graph, *i.e.,* $v_i = x_i$. As before, by N_j we denote the set of the K nearest neighbors of object x_i among $x_1, x_2, \cdots, x_{i-1}, x_{i+1}, \cdots, x_{|V|}$. The edges in G can be generated using the following rule: if $x_i \in N_j$ and $x_j \in N_i$, there is an edge between x_i and x_j.

4. Experiment

In order to illustrate the behavior of the proposed SELP algorithm, some experiments on classification tasks will be reported in this section. The semi-supervised community detection algorithm using label propagation (SLP) [5] and the unsupervised label propagation algorithm were used for comparison. In SELP, parameters α_0 and β were set to 1 and 2, respectively, as default. Parameter γ was determined by Eq. (21). Parameter η, which controls the prediction confidence, was set to be 0.7. The maximum iterative step was set to be 1000, and the percentage of labeled nodes for stopping the algorithm, $PercFul$, was set to be 0.9.

For the graph data, the NMI index, which measures the similarity between the planted partitions (ground truth) and the detected communities in the graph was used for comparison. The NMI of two partitions A and B for a graph with $|V|$ nodes, $\mathrm{NMI}(A, B)$, can be calculated by

$$\mathrm{NMI}(A,B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} M_{ij}\log\left(\frac{M_{ij}|V|}{M_{i\cdot}M_{\cdot j}}\right)}{\sum_{i=1}^{C_A} M_{i\cdot}\log\left(\frac{M_{i\cdot}}{|V|}\right)+\sum_{j=1}^{C_B} M_{\cdot j}\log\left(\frac{M_{\cdot j}}{|V|}\right)}, \tag{23}$$

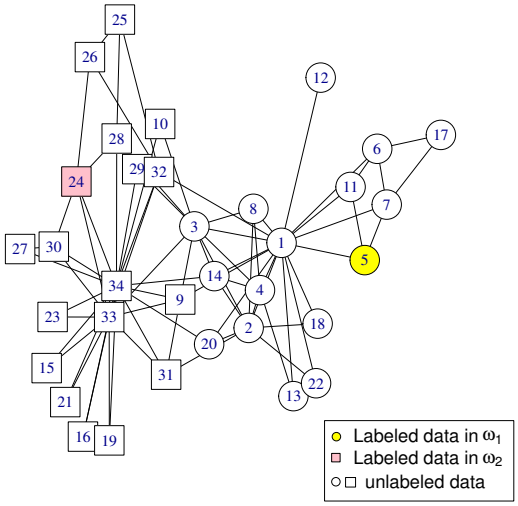
where C_A and C_B denote the numbers of communities in partitions A and B respectively. The notation M_{ij} stands for the element of matrix $(\boldsymbol{M})_{C_A\times C_B}$, representing the number of nodes in the i^{th} community of A that appears in the j^{th} community of B. The sum over row i of matrix $\boldsymbol{M}$ is denoted by $M_{i\cdot}$ and that over column j by $M_{\cdot j}$.

4.1. Graph data

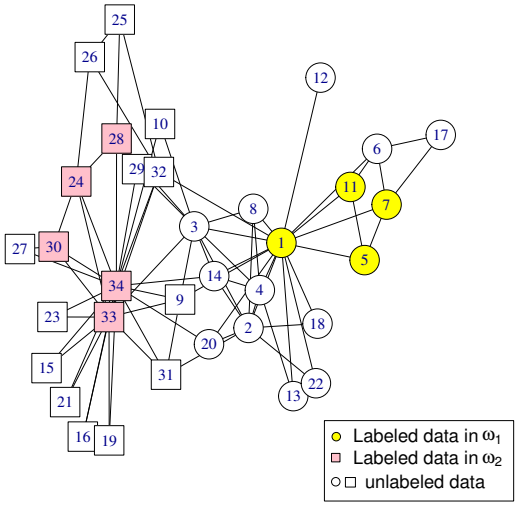
Example 1. Here we test on a widely used benchmark in detecting community structures, "Karate Club", studied by Wayne Zachary. The network consists of 34 nodes and 78 edges representing the friendship among the members of the club (see Figure 1-a). During the development, a dispute arose between the club's administrator and instructor, which eventually resulted in the club splitting into two smaller clubs. The first one is an instructor-centered group covering 16 vertices (the circle nodes in the figure), while the second administrator centered group consists of the remaining 18 vertices (the square nodes in the figure).

The labeled node in community ω_1 was set to node 5, while that in community ω_2 was node 24. After five steps, SELP algorithm stopped. The detailed update process is displayed in Figure 1. It can be seen from the figure that two outliers, nodes 10 and 12 are detected by SELP. From the original graph, we can see that node 10 has two neighbors, node 3 and node 34. But neither of them shares a common neighbor with node 10. For node 12, it only connects to node 1, but has no connection with any other node in the graph. Therefore, it is very intuitive that the two nodes are regarded as outliers of the graph.

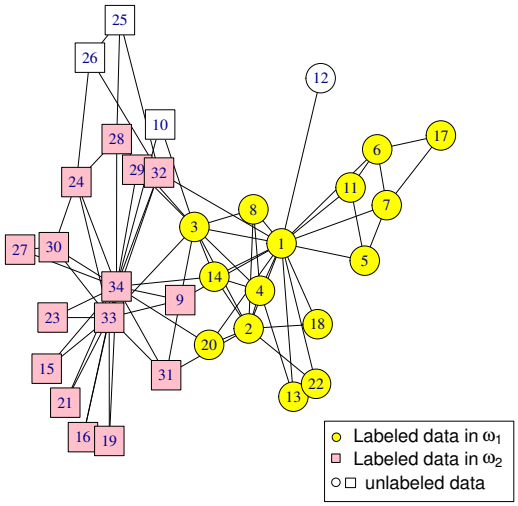
The detection results on Karate Club network by SELP and SLP algorithms with different labeled nodes are shown in Table 1. The labeled vertices and its corresponding misclassified



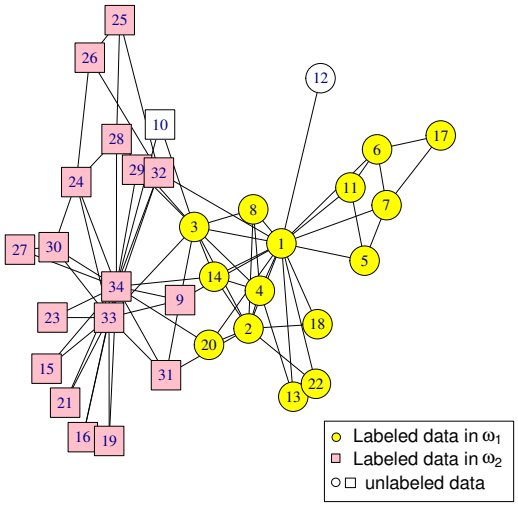
a. Initialization



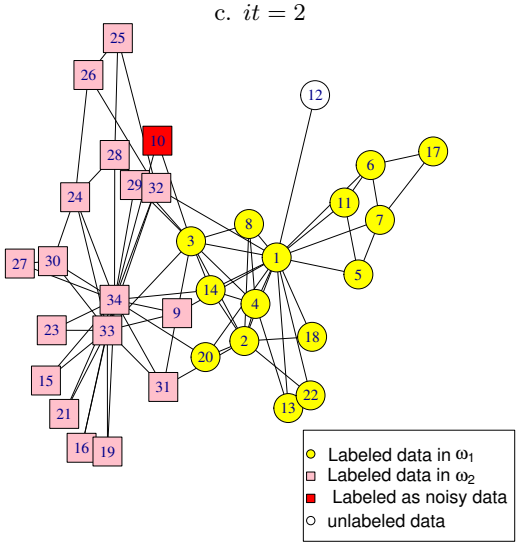
b. $it = 1$



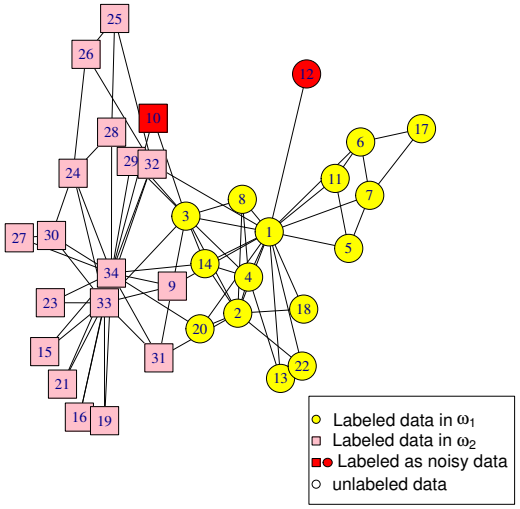
c. $it = 2$



d. $it = 3$



e. $it = 4$



f. $it = 5$

Figure 1: The label propagation process on Karate Club network. The nodes marked with color red are the outliers detected by SELP.

vertices are clearly presented. As it can be seen from the table, Nodes 10 and 12 are detected as outliers in all the cases by SELP, and the two communities can be correctly classified most of the time. The performance of SLP is worse than that of SELP when there is only one labeled data in each community. For the nodes which are connected to both communities and located in the overlap, such as nodes 3 and 9, they are misclassified most frequently. If the number of labeled data in each community is increased to 2, the exact community structure can be obtained by both methods. It is indicated that the more the prior information, *i.e.,* labeled vertices, the better the performance of SELP is.

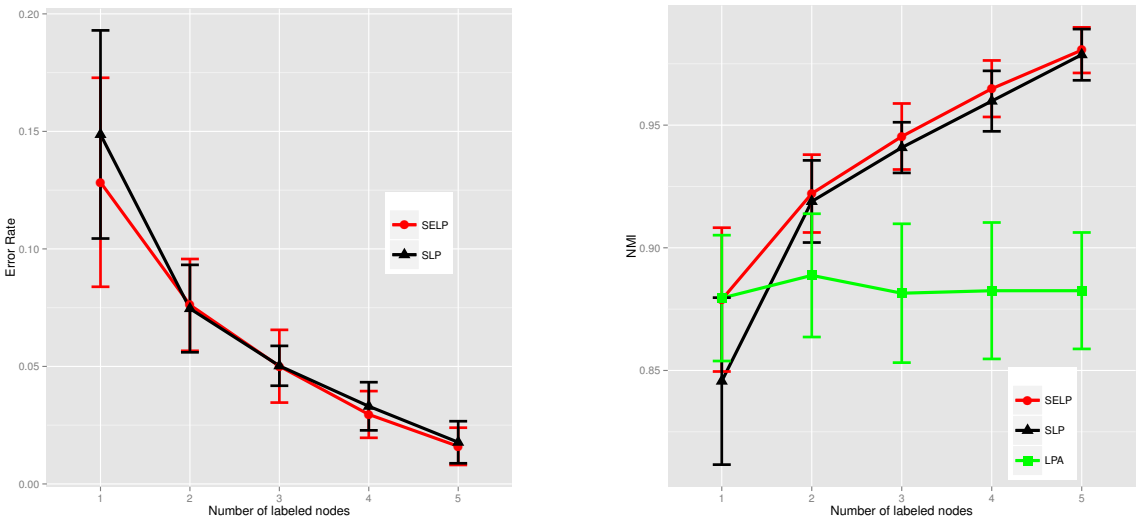
Table 1: Community detection results for the Karate Club network.

Labeled nodes in ω_1	Labeled nodes in ω_2	Misclassified nodes by SELP	Detected outliers by SELP	Misclassified nodes by SLP
1	34	None	10, 12	None
1	32	9	10, 12	9, 10, 27, 31, 34
2	33	None	10, 12	None
6	31	3	10, 12	2, 3, 8, 14, 2
8	31	None	10, 12	10
8	32	None	10, 12	None
17	31	3, 4, 8, 14	10, 12	2, 3, 4, 8, 13, 14, 18, 20, 22
1, 2	33, 34	None	10, 12	None
1, 2	33, 9	None	10, 12	None
3, 18	26, 30	None	10, 12	None
17, 4	31, 9	None	10, 12	None

Example 2. As a further test of our algorithm, the network we investigate in this experiment is the world of American college football games between Division IA colleges during regular season Fall 2000. The vertices in the network represent 115 teams, while the links denote 613 regular-season games between the two teams they connect. The teams are divided into 12 conferences containing around 8-12 teams each and generally games are more frequent between members from the same conference than between those from different conferences.

Let the number of labeled nodes in each community be fixed. Then SELP and SLP algorithms are run 50 times respectively with randomly selected labeled nodes. The average error rates and NMI values (plus and minus one standard deviation) of the 50 experiments are displayed in Figures 2-a and 2-b, respectively. As can be seen from the figures, with the increasing number of labeled samples, the performances of both SELP and SLP become better. The NMI values of the detected communities by SELP and SLP are significantly better than those by LPA. This finding indicates that the semi-supervised community detection methods could take advantage of the limited amount of prior information and consequently improve the accuracy of the detection results. The behavior of SELP is slightly better than that of SLP in terms of both error rates and NMI values.

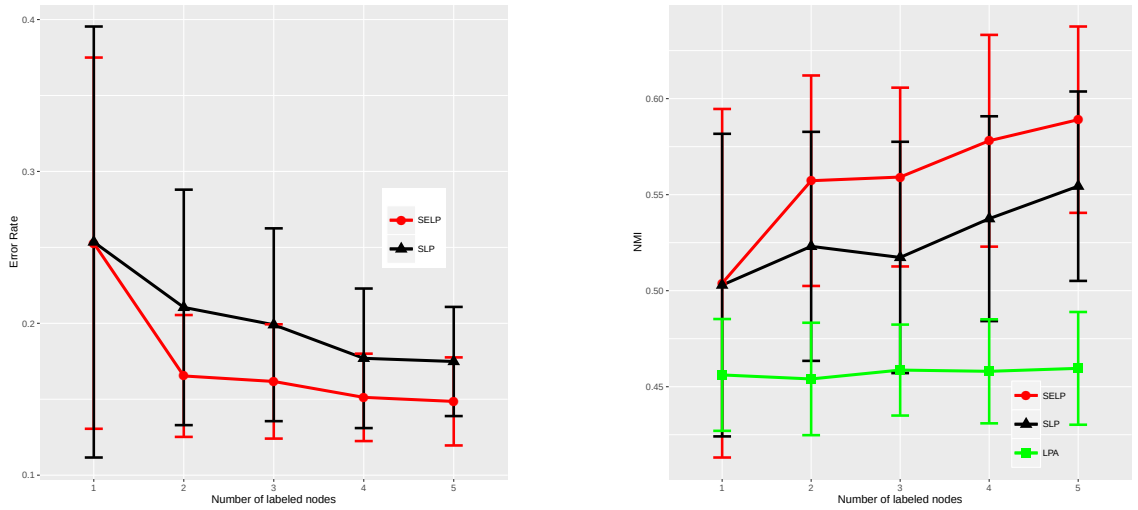
Example 3. In this example, we tested our algorithm on a large real-world network named com-Youtube, in which the ground-truth communities are known [37]. We selected 5637 nodes from 25 communities, which are in the top 5000 communities with highest quality that are described [37]. The selected communities contained at least five members. We performed some tests with different number of labeled nodes in each group. The results are displayed in Figure 3. From these figures, we can see that the detection results obtained by SELP and SLP are better than those of LPA, as the former two methods can take the limited supervised



a. Error rate b. NMI

Figure 2: The results on American football network.

information into consideration. SELP performs better than SLP in terms of the average values of the error rate and NMI index.



a. Error rate b. NMI

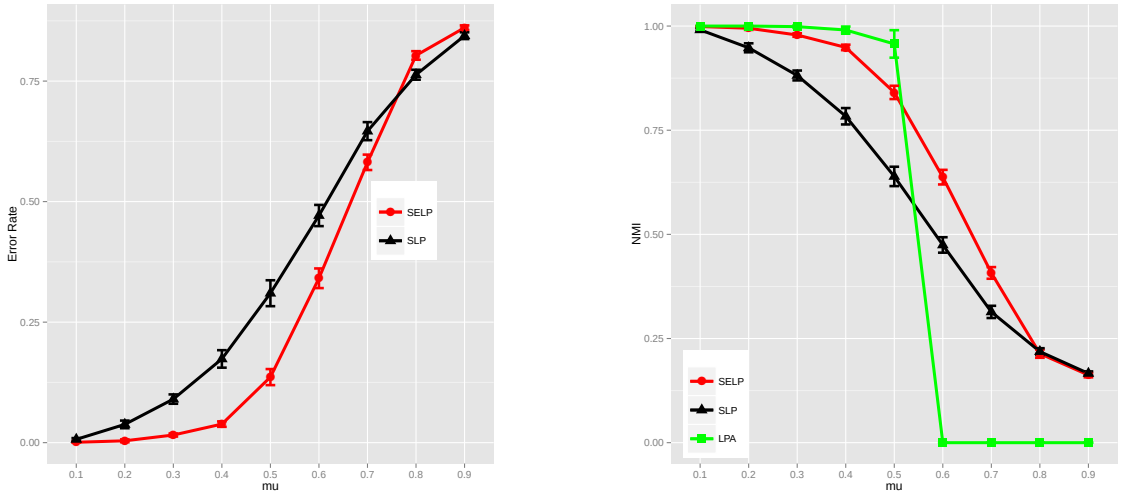
Figure 3: The results on Youtube network.

Example 4. In this experiment, we used a generated network frequently used for testing community detection approaches, the LFR benchmark [38], based on the assumption that the distributions of degree and community size are power laws. The experiments reported here included evaluating the performance of the algorithm with various amounts of labeled nodes and different values of parameter μ in the benchmark networks. The original LPA [4] and the semi-supervised community detection approach SLP [5] were used for comparison.

In LFR networks, the mixing parameter μ represents the ratio between the external degree of each vertex with respect to its community and the total degree of the node. The

larger the value of μ, the more difficult it is to detect the community structure. The values of the parameters in LFR benchmark networks were first set as follows: $n = 1000$, $\xi = 15$, $\tau_1 = 2$, $\tau_2 = 1$, $cmin = 20$, $cmax = 50$.

The performances of different methods with various values of μ are shown in Figure 4. As expected, the error rate is very high and the NMI value is low when μ is large. From Figure 4-b, we can see the original LPA could not work at all when μ is larger than 0.5. This result reflects the fact that the community structure is not very clear and consequently difficult to be identified correctly. It can be seen from Figure 4-a that the error rates of SELP are generally smaller than those of SLP. SELP performs better than SLP. The same conclusion can be drawn from the NMI values displayed in Figure 4-b.

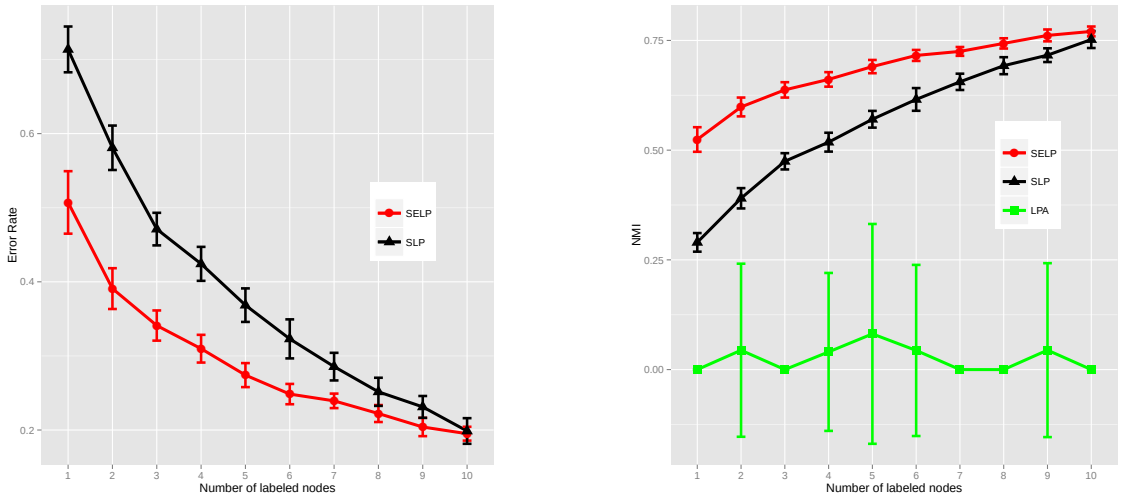


a. Error rate b. NMI

Figure 4: The results on LFR network with $n = 1000$. The number of labeled nodes in each community is 3.

We continue some experiments on LFR benchmarks with fixed parameter $\mu = 0.6$. Different numbers of labeled nodes in each community were adopted for SELP and SLP in the experiment. The results are displayed in Figure 5. As we can see, LPA become completely invalid as the NMI values of the detected community structure is around 0. The performance of SELP and SLP is significantly improved compared with LPA. As shown in Figure 5-b, even when there is only one labeled data in each community, the behavior of SELP is much better than that of LPA. This confirms the fact that the semi-supervised community detection approaches can effectively take advantage of the limited amount of labeled data. From the figure, we can also see that the performance of SELP and SLP becomes better with the increasing number of labeled nodes.

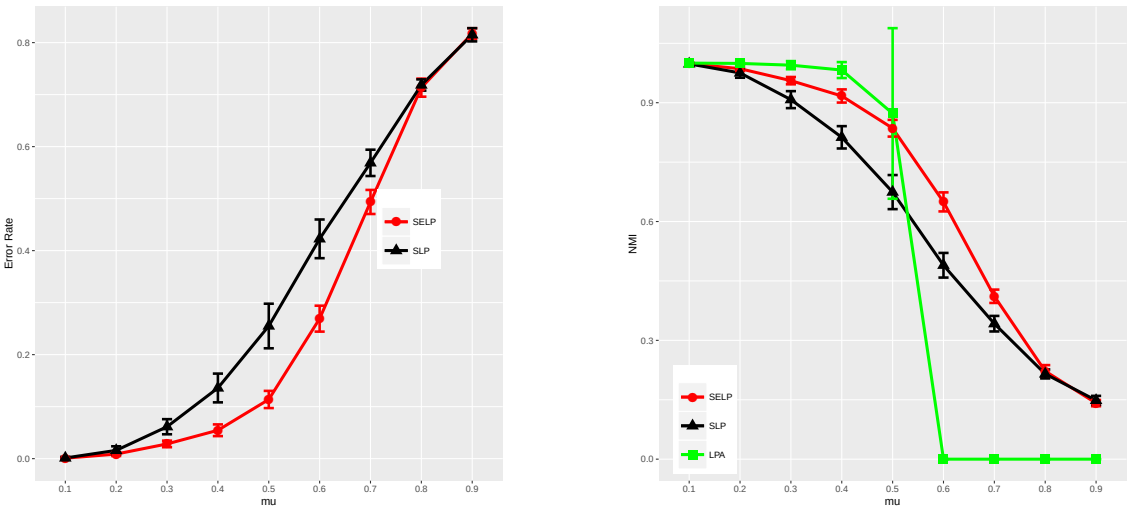
We also test on a large LFR benchmark with the following parameters: $n = 5000$, $\xi = 30$, $\tau_1 = 2$, $\tau_2 = 1$, $cmin = 500$, $cmax = 1000$. The results are displayed in Figures 6 and 7. As can been seen from Figure 6, SELP is superior to SLP and LPA, especially when μ is



a. Error rate b. NMI

Figure 5: The results on LFR network with $n = 1000$. The parameter of μ is set to be 0.6.

between 0.4 and 0.7. Figure 7 shows the results with different number of labeled nodes in each community. For both SELP and SLP, when there are more labeled nodes, the detection results become better in terms of error rate and NMI index. This is in consistent with our common sense. From Figure 7-b, we can see that LPA does not work at all when $\mu = 0.6$. But in this case SELP can also provide good results even when there is only one labeled node in each community.

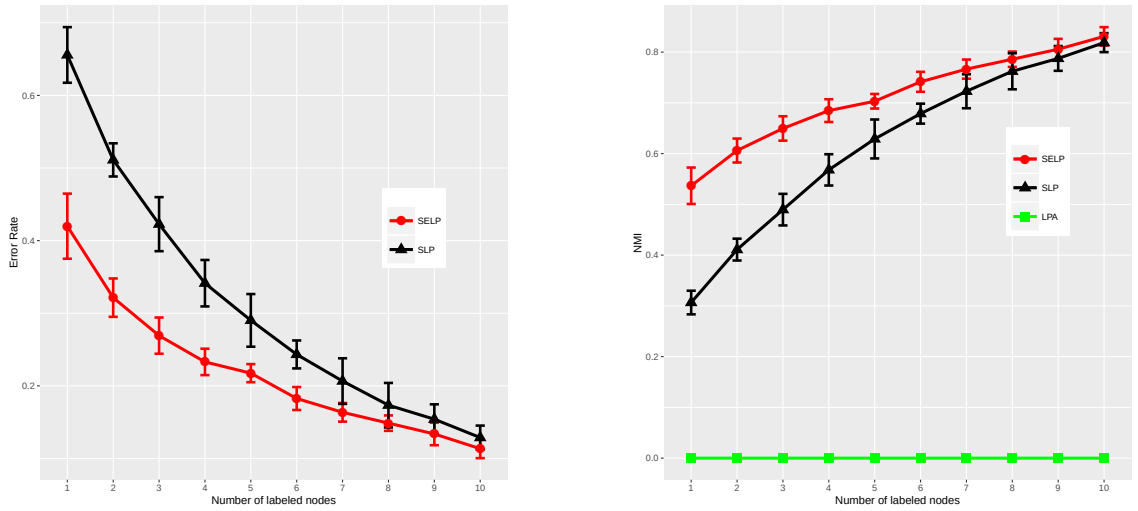


a. Error rate b. NMI

Figure 6: The results on LFR network with $n = 5000$. The number of labeled nodes in each community is 3.

4.2. Classical data sets

This section is to show some perspectives on the application of SELP on some classical data sets. The KNNG method was used to construct graphs based on the Euclidean distance

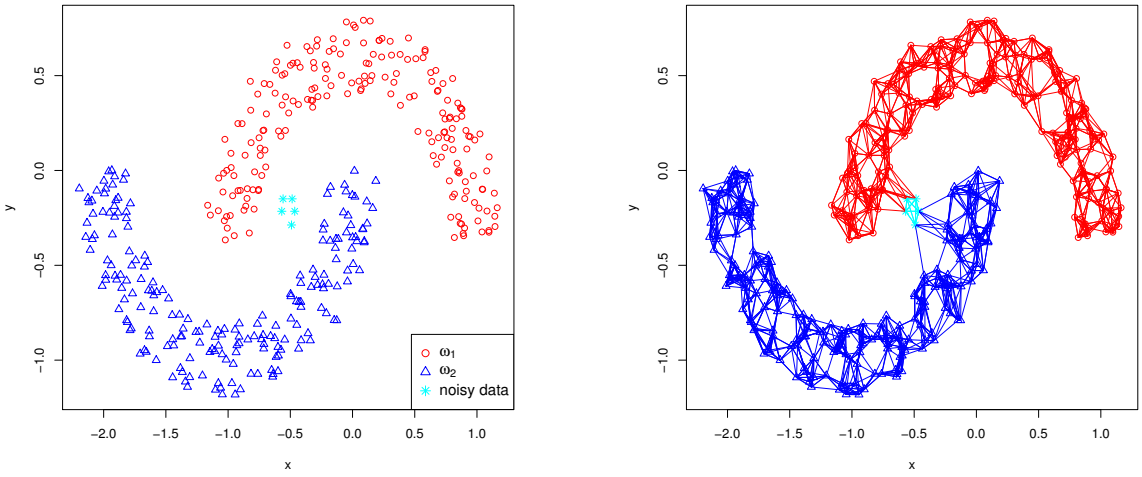


a. Error rate b. NMI

Figure 7: The results on LFR network with $n = 5000$. The parameter of μ is set to be 0.6.

between objects. We also compared our method with the K-EVCLUS clustering algorithms which is designed for relational data [29]. The dissimilarity measure adopted in K-EVCLUS was also the Euclidean distance.

Example 5. The performance of SELP will be first tested on a simulated two-dimensional data set which is shown in Figure 8-a. This data set consists of 405 objects which form two non-linearly separable semi-circle shaped clusters. There are five noisy data (marked with stars in the figure) that does not belong to either class. The constructed graph using the KNNG method with $K = 9$ is displayed in Figure 8-b.

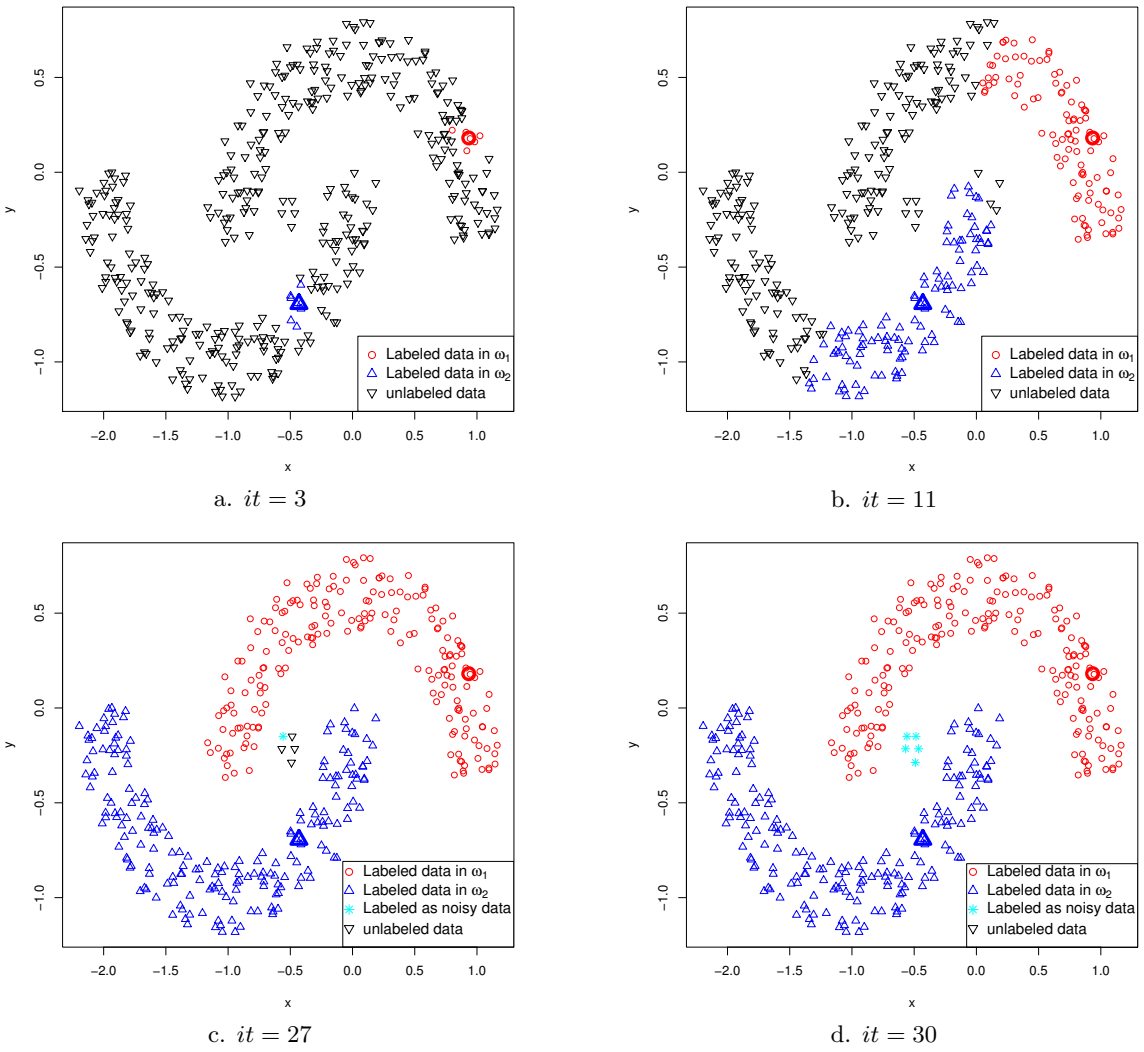


a. The original data set b. The constructed graph

Figure 8: The two-moon data set.

We ran SELP algorithm on this graph. Initially we randomly select one sample from each class as the labeled data. The label propagation process is illustrated in Figure 9. As it can be seen from the figure, the algorithm stops after 30 iterations. The two classes as well as the five noisy data are correctly classified finally. Using other semi-supervised learning algorithms

such as SLP on the same constructed graph, the two classes are easily detected. However, the noisy data will be partitioned into the two classes randomly. Setting $K = 2$, the clustering result by K-EVCLUS is shown in Figure 10. As can be seen, the performance of K-EVCLUS is not good. The objects located in the left part of the top moon and those in the right part of the bottom moon are not correctly classified. This can be due to the fact that K-EVCLUS does not make use of the supervised information, as it is an unsupervised clustering method.



a. $it = 3$ b. $it = 11$

c. $it = 27$ d. $it = 30$

Figure 9: The label propagation process on two-moon data set. The initial labeled samples are marked with big size symbols in the figure.

Example 6. The original data set used in the example is shown in Figure 11-a, which is a three-ring pattern with 180×3 data points. Each circle contains 180 points. There are also eight noisy points located between the circles. Figure 11-b depicts the constructed graph using the KNNG method with $K = 10$.

The update process and the classification results are illustrated in Figures 12-a–d. From these figures, we can see that SELP could detect the three classes exactly. Five outliers out of six are correctly found. The results by SLP are not shown here, as it can not provide good results due to the outliers in the data sets. We also tried K-EVCLUS by setting $K = 3$. The result is not satisfactory. Even in the same ring, objects are partitioned into different groups

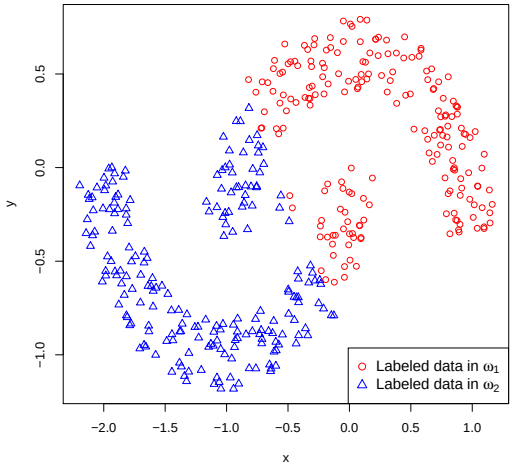
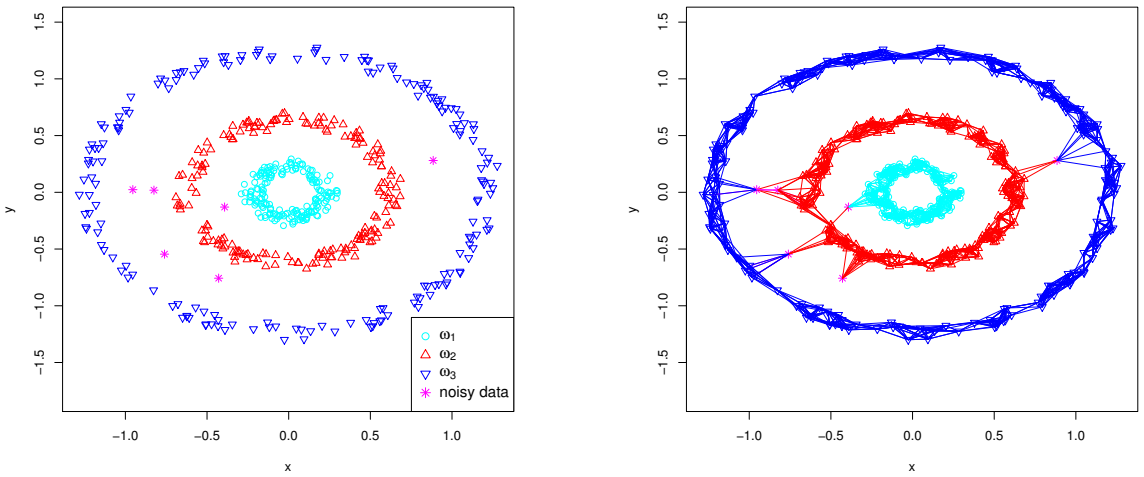


Figure 10: The clustering result on two-moon data set with K-EVCLUS clustering method.



a. The original data set

b. The constructed graph

Figure 11: The three-ring data set.

by K-EVCLUS. This fact confirms the advantage of limited supervised information, which can be used to improve the performance of the clustering algorithm.

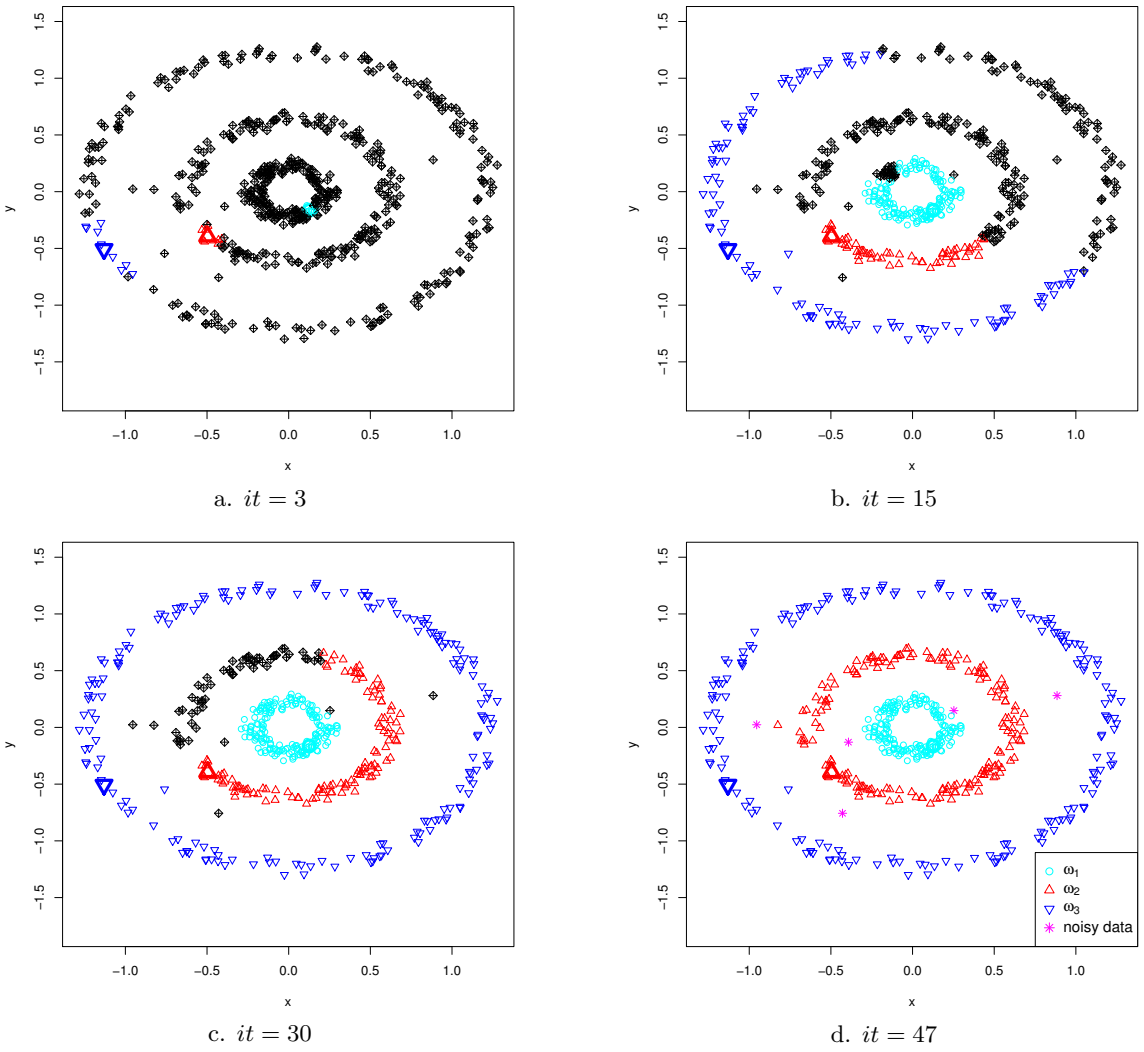


Figure 12: The label propagation process on three-ring data set. The initial labeled samples are marked with big size symbols in the figure. In the first three figures, the samples marked with black symbols denote the unlabeled data in the current step.

These two experiments on classical data sets are just used to show the possibility of the application of SELP on classical data sets. The results indicate that SELP works well especially in detecting outliers. However, how to construct the graph using the dissimilarities between objects needs a further consideration.

5. Conclusion

In this paper, the SELP algorithm has been introduced as an enhanced version of LPA. The approach proposed here can effectively take advantage of the limited amount of supervised information. This advantage is of practical meaning in real applications as there often exists some prior knowledge about the analyzed data sets. The experimental results show that the

detection results will be significantly improved with the help of limited amount of supervised data.

At the end of the experimental part, we showed the possibility to apply SELP on classical data sets. This task requires the construction of graph based on the dissimilarities between objects. However, how to construct graphs should be further studied. Another problem, which can be seen from the experiments, is that the detection results are different if the labeled data are different. Thus, which data should be selected as the initial labeled samples should be studied. This question is related to active learning. Active community detection using the principle of label propagation is also an interesting problem that is worth of investigation in future research work.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Nos.61701409, 61135001, 61403310, 61672431) and the Fundamental Research Funds for the Central Universities of China (No.3102016QD088).

References

[1] K. Zhou, A. Martin, Q. Pan, Semi-supervised evidential label propagation algorithm for graph data, in: International Conference on Belief Functions, Springer, 123–133, 2016.

[2] S. Fortunato, Community detection in graphs, Physics reports 486 (3) (2010) 75–174.

[3] S. Fortunato, D. Hric, Community detection in networks: A user guide, Physics Reports 659 (2016) 1–44.

[4] U. N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, Physical Review E 76 (3) (2007) 036106.

[5] D. Liu, H.-Y. Bai, H.-J. Li, W.-J. Wang, Semi-supervised community detection using label propagation, International Journal of Modern Physics B 28 (29) (2014) 1450208.

[6] L. Yang, X. Cao, D. Jin, X. Wang, D. Meng, A unified semi-supervised community detection framework using latent space graph regularization, IEEE transactions on cybernetics 45 (11) (2015) 2585–2598.

[7] X. Liu, W. Wang, D. He, P. Jiao, D. Jin, C. V. Cannistraci, Semi-supervised community detection based on non-negative matrix factorization with node popularity, Information Sciences 381 (2017) 304–321.

[8] T. Denœux, A k-nearest neighbor classification rule based on Dempster-Shafer theory, Systems, Man and Cybernetics, IEEE Transactions on 25 (5) (1995) 804–813.

[9] P. Xu, F. Davoine, H. Zha, T. Denoeux, Evidential calibration of binary SVM classifiers, International Journal of Approximate Reasoning 72 (2016) 55–70.

[10] C. Lian, S. Ruan, T. Denœux, Dissimilarity Metric Learning in the Belief Function Framework, IEEE Transactions on Fuzzy Systems 24 (6) (2016) 1555–1564.

[11] T. Reineking, Active classification using belief functions and information gain maximization, International Journal of Approximate Reasoning 72 (2016) 43–54.

[12] Z.-g. Liu, Q. Pan, J. Dezert, A. Martin, Adaptive imputation of missing values for incomplete pattern classification, Pattern Recognition 52 (2016) 85–95.

[13] M.-H. Masson, T. Denoeux, ECM: An evidential version of the fuzzy c-means algorithm, Pattern Recognition 41 (4) (2008) 1384–1397.

[14] T. Denœux, M.-H. Masson, EVCLUS: evidential clustering of proximity data, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 34 (1) (2004) 95–109.

[15] Z.-g. Liu, Q. Pan, J. Dezert, G. Mercier, Credal c-means clustering method based on belief functions, Knowledge-Based Systems 74 (2015) 119–132.

[16] K. Zhou, A. Martin, Q. Pan, Z.-G. Liu, ECMdd: Evidential c-medoids clustering with multiple prototypes, Pattern Recognition 60 (2016) 239–257.

[17] K. Zhou, A. Martin, Q. Pan, Evidential communities for complex networks, in: Information Processing and Management of Uncertainty in Knowledge-Based Systems, Springer, 557–566, 2014.

[18] K. Zhou, A. Martin, Q. Pan, Z.-g. Liu, Median evidential c-means algorithm and its application to community detection, Knowledge-Based Systems 74 (2015) 69–88.

[19] K. Zhou, A. Martin, Q. Pan, The belief noisy-or model applied to network reliability analysis, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 24 (06) (2016) 937–960.

[20] V.-D. Nguyen, V.-N. Huynh, Two-probabilities focused combination in recommender systems, International Journal of Approximate Reasoning 80 (2017) 225–238.

[21] Y. Tang, D. Zhou, S. Xu, Z. He, A Weighted Belief Entropy-Based Uncertainty Measure for Multi-Sensor Data Fusion, Sensors 17 (4) (2017) 928.

[22] E. Côme, L. Oukhellou, T. Denoeux, P. Aknin, Learning from partially supervised data using mixture models and belief functions, Pattern recognition 42 (3) (2009) 334–348.

[23] T. Denoeux, Maximum likelihood estimation from uncertain data in the belief function framework, Knowledge and Data Engineering, IEEE Transactions on 25 (1) (2013) 119–130.

[24] O. Kanjanatarakul, T. Denoeux, S. Sriboonchitta, Prediction of future observations using belief functions: A likelihood-based approach, International Journal of Approximate Reasoning 72 (2016) 71–94.

[25] P. Smets, Decision making in the TBM: the necessity of the pignistic transformation, International Journal of Approximate Reasoning 38 (2) (2005) 133–147.

[26] G. Shafer, A mathematical theory of evidence, Princeton University Press, 1976.

[27] P. Smets, Belief Functions: the Disjunctive Rule of Combination and the Generalized Bayesian Theorem, International Journal of Approximate Reasoning 9 (1993) 1–35.

[28] S. B. Hariz, Z. Elouedi, K. Mellouli, Clustering approach using belief function theory, in: International Conference on Artificial Intelligence: Methodology, Systems, and Applications, Springer, 162–171, 2006.

[29] T. Denoeux, S. Sriboonchitta, O. Kanjanatarakul, Evidential clustering of large dissimilarity data, Knowledge-Based Systems 106 (2016) 179–195.

[30] T. Denœux, O. Kanjanatarakul, Evidential Clustering: A Review, in: Integrated Uncertainty in Knowledge Modelling and Decision Making: 5th International Symposium, IUKM 2016, Da Nang, Vietnam, November 30-December 2, 2016, Proceedings 5, Springer, 24–35, 2016.

[31] T. Denœux, O. Kanjanatarakul, S. Sriboonchitta, EK-NNclus: A clustering procedure based on the evidential K-nearest neighbor rule, Knowledge-Based Systems 88 (2015) 57–69.

[32] X. Zhu, J. Lafferty, R. Rosenfeld, Semi-supervised learning with graphs, Carnegie Mellon University, language technologies institute, school of computer science, 2005.

[33] M. Li, Z.-H. Zhou, SETRED: Self-training with editing, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 611–621, 2005.

[34] X. Zhu, Semi-supervised learning literature survey, Tech. Rep., Computer Sciences Technical Report 1530, University of Wisconsin, Madison, 2006.

[35] F. Wang, C. Zhang, Label propagation through linear neighborhoods, IEEE Transactions on Knowledge and Data Engineering 20 (1) (2008) 55–67.

[36] M. Maier, M. Hein, U. von Luxburg, Optimal construction of k-nearest-neighbor graphs for identifying noisy clusters, Theoretical Computer Science 410 (19) (2009) 1749–1764.

[37] J. Yang, J. Leskovec, Defining and evaluating network communities based on ground-truth, Knowledge and Information Systems 42 (1) (2015) 181–213.

[38] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Physical review E 78 (4) (2008) 046110.